

# Design of Artificial Neural Networks Based on Genetic Algorithms to Forecast Time Series

Juan Peralta, German Gutierrez, and Araceli Sanchis



CAOS Group, Computer Science Department, University Carlos III of Madrid  
{jperalta,ggutierr,masm}@inf.uc3m.es

**Abstract.** In this work an initial approach to design Artificial Neural Networks to forecast time series is tackled, and the automatic process to design is carried out by a Genetic Algorithm. A key issue for these kinds of approaches is what information is included in the chromosome that represents an Artificial Neural Network. There are two principal ideas about this question: first, the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network, i.e. Direct Encoding Scheme; second, the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture), i.e. Indirect Encoding Scheme. The results for a Direct Encoding Scheme (in order to compare with Indirect Encoding Schemes developed in future works) to design Artificial Neural Networks for NN3 Forecasting Time Series Competition are shown.

**Keywords:** Time Series Forecasting, Artificial Neural Networks Design, Genetic Algorithms, Direct Encoding Scheme.

## 1 Introduction

In order to acquire knowledge, it is so interesting to know what the future will look like, i.e. forecast the future from the observed past. The forecasting task can be performed by several techniques as Statistical methods, Immune Systems, and Artificial Neural Networks (ANN).

This contribution reports the methodology to carry out the automatic design of ANN that tackles the Forecasting Time Series problem taken from NN3 Forecasting Time Series Competition [1].

NN3 suggests a forecasting time series competition to evaluate automatic methodologies for designing ANN by means of several time series. Two important research questions are pointed out: the performance of ANN compared with others forecasting methods; and “best practice” methodologies carried out by researchers to model ANN for time series forecasting. The task in NN3 Forecasting Competition is to forecast several time series, not all of them with the same ANN, but the participants have to develop just one methodology to obtain a different ANN to forecast each time series.

The first step in order to design an ANN is setting the kind of ANN with which to solve the forecasting task, and the learning algorithm used. According to [2], that shows the approximation capability of Multilayer Perceptron (MLP), we have focused

on MLP with only one hidden layer and Backpropagation (BP) as learning algorithm to forecasting time series.

The design of ANN consists in setting its parameter values. In the case of MLP with only one hidden layer and BP the parameters are: number of inputs nodes, number of hidden neurons (number of output neurons is placed by the problem), which is the connection pattern (how the nodes are connected), the whole set of connection weights (or learning rate, minimum error to reach and the maximum training cycles as parameters that allow together with BP obtain it).

The process of designing ANN could be considered as a search problem within all possible designs. And this search could be done by Genetic Algorithms (GA).

Several works approach the design of ANN using Evolutionary Techniques. Some of them use Direct Encoding Schemes (DES) [3,4], the others using Indirect Encoding Scheme (IES) [5,6,7,8,9]. For DES the chromosome contains information about parameters of the topology, architecture, learning parameters, etc. of the Artificial Neural Network. In IES the chromosome contains the necessary information so that a constructive method gives rise to an Artificial Neural Network topology (or architecture). In [10] evidence about performance of ANN obtained from GA are reported. Ajith Abraham [11] shows an automatic framework for optimization ANN in an adaptive way, and Xin Yao et. al. [12] try to spell out the future trends of the field.

The work reported in this contribution is based on a “Research Paper” of first author (and supervised by the later authors) for Master in Computer Science and Technology at Carlos III of Madrid University<sup>1</sup>.

The final object of our approach is consider Sparsely Connected MLP to forecast time series, and use both Indirect Encoding Scheme, one based on Cellular Automata [8] and other based on Bidimensional Grammar [9], to design ANN using GA. A previous work for designing ANN to forecast using GA based on Cellular Automata is developed by Salah and Al-Salqan in [13].

The paper is organized as follows. Sec 2 reviews questions about how to tackle forecast task with ANNs. Sec 3 approaches design of ANN to forecast with GA. In Sec 4 experimental setup and results are shown. And finally, conclusions and future works are described in Sec 5.

## 2 Forecasting Time Series with ANN

There are several works that tackles the forecasting time series task with ANN, not only computer science researchers, but statisticians as well. This reveals the full consideration of ANN (as a data driven learning machine) into forecasting theory.

Before using an ANN to forecast, it has to be designed, i.e. establishing the suitable value for each degree of freedom of the ANN [14] (kind of net, number of input nodes, number of outputs neurons, number of hidden layer, number of hidden neurons, the connections from one node to another, etc.). The design process is more an “art” based on test and error and the experience of human designer, than an algorithm. In [15] Crone proposed an “extensive modeling approach” to review

---

<sup>1</sup> <http://www.uc3m.es/uc3m/gral/TC/ESMAOF/CTII/ctii.html>

several designs of ANNs, and finally Crone and Preßmar show in [16] an evaluation framework to NN modeling in forecasting, and literature review.

In order that a single ANN could work, and forecasting, with time series values, an initial step from original values of time series have to be done, i.e. normalize the data. And, once the ANN gives those values, the inverse process is done. This step is important as the ANN will learn just the normalized values.

The problem of forecasting time series with ANN is considered as obtaining the relationship of the value of period "t" (i.e. the net has only one output neuron) and the values of previous periods, i.e the function (1):

$$a_t = f(a_{t-1}, a_{t-2}, \dots, a_{t-k}). \quad (1)$$

Therefore, the time series will be transform into a pattern set, it depend on the k inputs nodes of a particular ANN. Each pattern consists in:

- "k" inputs values, that correspond to "k" normalized previous values of period t:  $a_{t-1}, a_{t-2}, \dots, a_{t-k}$
- One output value, that corresponds to normalized time series value of period t.

The complete patterns set are ordered into the same way the time series is. This patterns set will be used to train and test the ANN, then it will be split into two sets, train and test sets. The train set will be obtained from the first  $m\%$  (e.g 70%) and the test set will be obtained from the rest of the complete patterns set.

If hand design of ANN is carry out, several topologies (i.e. different number of inputs nodes and number of different number of neurons of only one hidden layer), with different learning rates are trained. For each of them, train and test error are obtained, and one with better generalization capability (i.e. less test error and a good train error) is selected to generate forecast values.

### 3 ANN Design with Genetic Algorithms

The problem of design ANN could be seen as a search problem into the space of all possible ANN. And that search can be done by a GA [17] using exploitation and exploration. There three crucial issues: the solution's space; how each solutions is codified into a chromosome, i.e. Encoding Scheme; and what is looking for, translated into the function fitness.

As a first approach to design ANN to forecasting time series, a Direct Encoding Scheme for Full Connected MLP has been considered.

For this Direct Encoding Scheme the information placed into the chromosome is:

- Number of inputs nodes ( $i$ ).
- Number of hidden nodes ( $h$ ).
- Learning rate, for BP learning algorithm ( $\alpha$ )

The value of learning rate, " $\alpha$ ", is between 0 and 1, and the value of " $i$ " and " $h$ " is limited by a maximum number of inputs nodes ( $max\_inputs$ ) and a maximum number

of hidden nodes (*max\_hidden*), respectively. These maximum values are related, by a factor "a", with the number of periods of time series what values are known (*nts*), see ec. (2)

$$\begin{aligned} \max\_inputs &= a \times nts . \\ \max\_hidden &= 2 \times \max\_inputs . \\ a &= 0.3 . \end{aligned} \quad (2)$$

Into the chromosome, two decimal digits, i.e two genes, are used to codify the value "i", other two for "h" and the last two for "a". This way, the values of "i", "h" and "a" are obtained, from the chromosome, as it can see in (3)

$$\begin{aligned} \text{chrom: } &g_{i1} \ g_{i2} \ g_{h1} \ g_{h2} \ g_{a1} \ g_{a2} . \\ i &= \max\_inputs \times ((g_{i1} \cdot 10 + g_{i2}) / 100) . \\ h &= \max\_hiddens \times ((g_{h1} \cdot 10 + g_{h2}) / 100) . \\ a &= ((g_{a1} \cdot 10 + g_{a2}) / 100) . \end{aligned} \quad (3)$$

The search process (GA) begins with a random population, i.e set of randomly generated chromosomes. Later, the fitness value for each one of the individual of the population is obtained (a). Once that it is already done the GA operators as Elitism, Selection, Crossover and Mutation are applied in order to generate the population of next generation, i.e. set of chromosomes (b). The steps (a) and (b) are iteratively executed till a maximum number of generations is reached.

To obtained the fitness value of a chromosome:

1. The train patterns and test patterns sets are obtained, depending on the number of inputs nodes of the net, as was said above (Sec.2).
2. Then, the connection weights are randomly initialized, and the net is trained (using Stuttgart Neural Network Simulator (SNNS) binary tools [19]) a maximum training cycles.

The fitness value will be the minimum error test reached in training process; it doesn't have to be in the last training cycle. The architecture of the net (topology + connections weights) when the test error is minimum in training process is saved to be used later for forecasting time series values.

Once that GA reaches the last generation, the best individual from all generations is used to forecast the time series.

## 4 Experimental Setup and Results

The experiments related to our approach have been carried out on the Reduced Database of NN3 Competition. Nowadays, the results of time series, i.e. the time series values to be forecasted still have not been published till the moment. Therefore, the forecasted values obtained by our system cannot be evaluated or compared with real values. Nevertheless, the forecast results for ANN will be shown together forecast obtained with statistical techniques.

To normalize the time series values (Sec 2), so the nets will be able to work with them, it has to be considered the shape that time series has [16], specially trend (upward or downward) component. As the time series values have to be rescale, into

the numerical range value  $[0,1]$ , considering not only the known values, but the future values (those to be forecasted), the maximum and minimum limits for normalizing ( $max2norm$ ,  $min2norm$  respectively) cannot be just the maximum ( $max$ ) and minimum( $min$ ) known time series values. A margin from  $max$  and  $min$  has to be set if future values are higher or lower than they already are. Equation (4) show how are obtained  $max2norm$  and  $min2norm$ .

$$\begin{aligned} max2norm &= max + (0.1 \times (max-min)) . \\ min2norm &= min - (0.1 \times (max-min)) . \end{aligned} \quad (4)$$

The parameters of GA described in Sec 3 are: population size, 50; maximum number of generations, 100; percentage of the best individual that stay unchangeable to the next generation (percentage of elitism), 10%; crossover method, mutation probability,  $(1/length\_chrom) = 0.17$ .

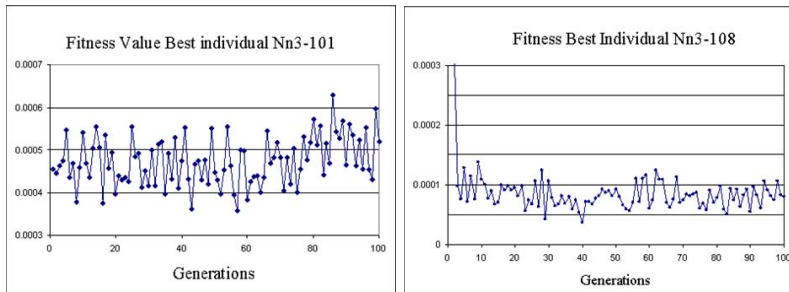
The parameter for learning process and getting fitness values are: percentage from whole pattern set to obtain train and test subsets (Sec 2), 70 % and 30% respectively; and maximum training cycles,  $10^4$ . Remember that number of inputs nodes, hidden neurons and learning rate values are obtained from chromosome.

And, as it was said before, the fitness function is the minimum error test reached in training process (Sec 3). At the end of this process, the topology obtained will be the one described by the information inside the chromosome with the best fitness, plus the connection weights obtained at the end of the training process of the net.

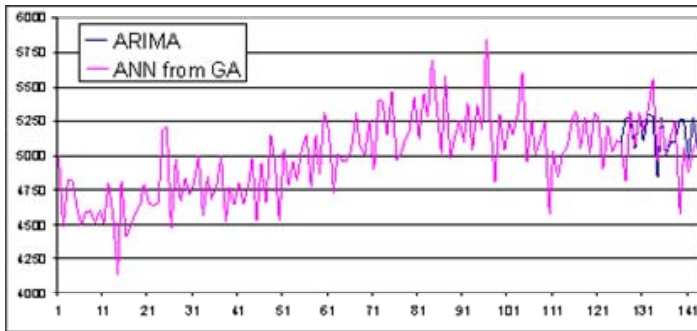
#### 4.1 Results of Genetic Algorithm

The GA to design an ANN has been carried out for the 11 time series of NN3 Reduced Database. Nevertheless, because of the limited space, the results for only two of them will be shown, those that seem especially significant, nn3-101 and nn3-108 time series.

For each of them, it will be shown on one side the evolution fitness graph along the generations (Fig 1), and on the other hand the known time series values together with the forecasted ones obtained by two methods (Fig 2 & 3). The two methods are: our GA method and a Statistical one (ARIMA [18], implemented in AUTOBOX tool).



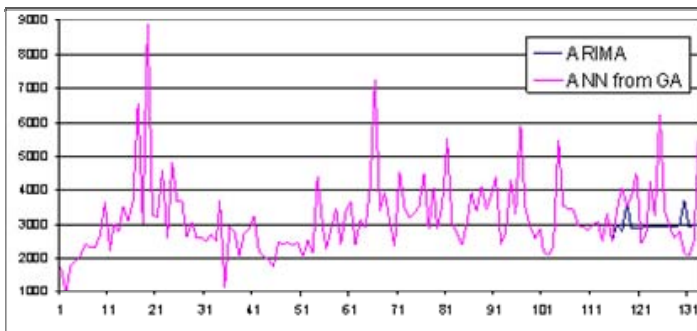
**Fig. 1.** Fitness along generations for nn3-101 and nn3-108



**Fig. 2.** Forecasted values for nn3-101

For our method, after GA runs, the best ANN model is selected and used to forecast just the 18 values asked by the competition.

As far as we check, the GA for nn3-101 does not improve the fitness value along 100 generations, i.e. GA is not better than a random search for this time series and these GA parameters. However, for nn3-108 GA decreases the error test from the beginning. On the other hand it is interesting to emphasize the saw shape which has the graph of the fitness evolution (Fig 1), this happens because in each fitness evaluation of an individual along the GA the weights are randomly initialized for learning process. The set of connection weights are not included into the chromosome, though the search process generates an individual not only with an appropriate topology but also with a right set of free parameters (i.e. set of connection weights).



**Fig. 3.** Forecasted values for nn3-108

In relation to forecasting, it seems that, for both time series, with ANN obtained by GA goes on with the trend of time series.

## 5 Conclusions and Future Works

As we said before the set of connection weights are not included into the chromosome, though the search process generates an individual not only with an appropriate topology but also with a right set of free parameters (i.e. set of connection weights).

Furthermore, the Partial Autocorrelation Function (PACF) [18] indicates that the optimal value of previous elements of the Time Series (i.e. input nodes to be considered for ANN) to forecast a new one is 15. Nevertheless, the results of GA are different for both Time Series, nn3-101 and nn3-108, being the optimal previous values (i.e. input nodes) 30 and 34 respectively to forecast a new one.

The system reported here is a first approximation with Direct Encoding to design ANN. In Future works our approach will consider, not full connected nets, but Sparsely Connected MLP to obtain a better forecasting, using both Indirect Encoding Scheme, one based on Cellular Automata [8] and other based on Bidimensional Grammar [9], to design ANN using GA.

## References

1. 2006/07 Forecasting Competition for Neural Networks & Computational Intelligence. <http://www.neural-forecasting-competition.com/>
2. G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2, 303-314, 1989
3. T. Ash. Dynamic Node Creation in Backpropagation Networks ICS Report 8901, The Institute for Cognitive Science, University of California, San Diego (Saiensu-sh, 1988), 1988
4. D.B. Fogel, Fogel L.J. and Porto V.W. Evolving Neural Network, *Biological Cybernetics*, 63, 487-493, 1990.
5. Gruau F. "Genetic Synthesis of Boolean Neural Networks with a Cell Rewriting Developmental Process". *Proceedings of COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pp. 55-74, IEEE Computer Society Press, 1992.
6. Yao, X. and Lin, Y. A new evolutionary system for evolving artificial neural networks, *Transactions on Neural Networks*, 8(3): 694-713, 1997.
7. Kitano, H.: Designing Neural Networks using Genetic Algorithms with Graph Generation System, *Complex Systems*, 4, 461-476, 1990.
8. G. Gutiérrez, A. Sanchis, P. Isasi, J.M. Molina and I. M. Galván. Non-direct Encoding method based on Cellular Automata to design Neural Network Architectures. *Computer and Informatics*, 24 2005, No3.
9. Studying the Capacity of Grammatical Encoding to Generate FNN Architectures. Germán Gutiérrez, Beatriz Garcá, Jos  M. Molina and Araceli Sanchis. *Computational Methods in Neural Modeling, Lecture Notes in Computer Science, Volume 2686/2003. IWANN 2003*.
10. James V Hansen, James B McDonald, Ray D Nelson. Time Series Prediction With Genetic-Algorithm Designed Neural Networks: An Empirical Comparison With Modern Statistical Models. *Computational Intelligence Journal*, Vol. 15, 171-184, 1999.
11. Ajith Abraham, Meta-Learning Evolutionary Artificial Neural Networks, *Neurocomputing Journal, Elsevier Science, Netherlands*, Vol. 56c, pp. 1-38, 2004.

12. H. Garis, D. Fogel, M. Conrad and X. Yao. Special Issue on evolutionary neural systems. *Neurocomputing*, 42, 1-8, 2002
13. Asma Abu Salah, Y.A. Meta-Learning Evolutionary Artificial Neural Networks Using Cellular Configurations: *Experimental Works Intelligent Computing*, 2006, Volume 4113/2006, pages 178-193
14. Haykin, S. Simon & Schuster (ed.) *Neural Networks. A Comprehensive Foundation* Prentice Hall, 1999
15. Crone, S. F. Stepwise Selection of Artificial Neural Networks Models for Time Series Prediction *Journal of Intelligent Systems*, Department of Management Science Lancaster University Management School Lancaster, United Kingdom, 2005,
16. Crone, S. F. and Preßmar, D. B. 2006. An extended evaluation framework for neural network publications in sales forecasting. *Proceeding of International Conference on Artificial intelligence and Applications*, 2006
17. Fogel, D. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley-IEEE Press, 1998.
18. ARIMA and PACF: <http://www.statsoft.com/textbook/sttimser.html>
19. Prof. Dr. Andreas Zell, WSI Computer Science Department, Computer Architecture, Software, Artificial Neural Networks <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

## Appendix

SNNS binary tools ( `ff_bignet` and `batchman`) [19], has been used for generate and training ANN. The whole source code is available in `{ggutierr,jperalta}@inf.uc3m.es`.